OpenBSD

Daniel Plakosh, Software Engineeering Institute [vita[1]]

2005-09-27

The OpenBSD UNIX variant was designed with an additional emphasis on security. In particular, OpenBSD adopted phkmalloc and adapted it to support guard pages and randomization.

Development Context

Dynamic memory management

Technology Context

C++, C, OpenBSD

Attacks

Attacker executes arbitrary code on machine with permissions of compromised process or changes the behavior of the program.

Risk

Standard C dynamic memory management functions such as `malloc()`, `calloc()`, and `free()` [ISO/IEC 99] are prone to programmer mistakes that can lead to vulnerabilities resulting from buffer overflow in the heap, writing to already freed memory, and freeing the same memory multiple times (e.g., double-free vulnerabilities).

Description

The OpenBSD UNIX variant was designed with an additional emphasis on security. OpenBSD adopted phkmalloc and adapted it to support guard pages and randomization. Table 1 shows some of the additional security options added for the OpenBSD version of phkmalloc. The default options are AJ.

Table 1. OpenBSD additional phkmalloc options

Flag	Description
F	“Freeguard.” Enable use after free protection. Unused pages on the freelist are read and write protected to cause a segmentation fault upon access.
G	“Guard.” Enable guard pages and chunk

[1] daisy:268 (Plakosh, Daniel)

	randomization. Each page size or larger allocation is followed by a guard page that will cause a segmentation fault upon any access. Smaller than page-size chunks are returned in a random order.

References

[ISO/IEC 99] ISO/IEC. *ISO/IEC 9899 Second edition 1999-12-01 Programming Languages — C.* International Organization for Standardization, 1999.

Pearson Education, Inc. Copyright

Velden

Naam	Waarde
Copyright Holder	Pearson Education

Velden

Naam	Waarde
is-content-area-overview	false
Content Areas	Knowledge/Coding Practices
SDLC Relevance	Implementation
Workflow State	Publishable